APPLICATION FOR UNITED STATES PATENT

in the name of

Kurt Frieden
Prasanna Srikhanta

of

Plumtree Software, Inc.

for

EFFICIENTLY STORING INDENTED THREADS IN A THREADED DISCUSSION APPLICATION

Law Office of Richard A. Dunning, Jr.
325M Sharon Park Drive, Suite 208
Menlo Park, CA 94025
Tel.: 650.234.1054
Fax: 650.234.8427

ATTORNEY DOCKET:	**DATE OF DEPOSIT:**	2/28/02
PLM012001	**EXPRESS MAIL NO.:**	ET161005857US

EFFICIENTLY STORING INDENTED THREADS IN A THREADED DISCUSSION APPLICATION

BACKGROUND

[0001] The present invention relates generally to data management, and particularly to managing data stored hierarchically.

[0002] Hierarchical data sets are useful and popular. Such data sets are generally composed of multiple records. In a hierarchical set of records, a record can have multiple "children," which are related to the record and exist at a lower level of the hierarchy. Such a record is referred to as the "parent" record of the children records. The children of a record may have children, and so on, limited only by the size of the database. Within a set of children of a single record, it is often useful to order the children, such that there is a first child, a second child, and so on.

[0003] To date the utility of these hierarchical data sets has been limited by the efficiency of the tools available for managing the data. Conventional methods of storing hierarchical data in a database involve multiple expensive calls to update and/or query the database. For example, in order to select the second grandchild of a record, three queries are necessary. The first query selects the child of the record. The second query selects the first child of the child. The third query selects the second child of the child.

SUMMARY

[0004] In general, in one aspect, the invention features a method, apparatus, and computer-readable media for retrieving records in a hierarchical set of the records having a plurality of hierarchical levels and a plurality of hierarchical depths, each of the records having a tag that is unique within the hierarchical set of the records. It comprises identifying one of the records in the hierarchical set of the records; modifying the tag, thereby producing a key; indexing the hierarchical set of the records only once, thereby selecting one or more of the records within the hierarchical set of the records, wherein indexing the hierarchical set of the records only once comprises applying the key to the hierarchical set of the records; and retrieving the selected records.

[0005] Particular implementations can include one or more of the following features. Applying comprises selecting those of the records in the hierarchical set of the

records having a tag that matches the key. Identifying one of the records comprises receiving a selection of the one of the records from a user; and receiving a command from the user; and wherein modifying the tag is based on the command from the user. Each of the records has one or more fields, and implementations can comprise displaying a field of each of the retrieved records on a display, wherein the position of each of the fields on the display represents the hierarchical depth and hierarchical level of the corresponding one of the retrieved records. Each tag is a number having a plurality of digits; the position of each of the digits represents one of the hierarchical depths; the value of each of the digits represents one of the hierarchical levels; and modifying the tag comprises selecting at least one of the digits according to the command from the user; and changing the value of the selected digits according to the command from the user. Each tag is a number having a plurality of digits; the position of each of the digits represents one of the hierarchical depths; the value of each of the digits represents one of the hierarchical levels; the command from the user requests retrieving the children of the identified record; and modifying the tag comprises selecting the digit corresponding to the hierarchical depth of the identified record; and setting the value of each digit corresponding to a hierarchical depth below the hierarchical depth corresponding to the selected digit to a wildcard value. Each tag is a number having a plurality of digits; the position of each of the digits represents one of the hierarchical depths; the value of each of the digits represents one of the hierarchical levels; the command from the user requests retrieving the parent of the identified record; and modifying the tag comprises selecting the digit corresponding to the hierarchical depth of the identified record; and setting the value of the selected digit to a null value. Each of the records represents one of a message and a folder.

[0006] In general, in one aspect, the invention features a method, apparatus, and computer-readable media for adding a new record to a hierarchical set of records having a plurality of hierarchical levels and a plurality of hierarchical depths, each of the records in the hierarchical set of records having a tag that is unique within the hierarchical set of records. It comprises identifying one of the records in the hierarchical set of records as the parent of the new record; modifying the tag, thereby producing a key; adding the key to the new record; and indexing the hierarchical set of records only once, thereby adding the new record to the hierarchical set of records, wherein indexing the hierarchical set of records only once comprises applying the key to the hierarchical set of records.

[0007] Particular implementations can include one or more of the following features. Identifying one of the records comprises receiving a selection of the one of the records from a user. Each tag is a number having a plurality of digits; the position of each of the digits represents one of the hierarchical depths; the value of each of the digits represents one of the hierarchical levels; the identified record represents a message; identifying one of the records further comprises receiving a command from the user that requests replying to the message; and modifying the tag comprises selecting the digit corresponding to the hierarchical depth immediately below the hierarchical depth of the identified record; and incrementing the value of the selected digit. Applying comprises selecting those of the records in the hierarchical set of the records having a tag that matches the key. Each tag includes a plurality of digits; the position of each of the digits represents one of the hierarchical depths; and the value of each of the digits represents one of the hierarchical levels.

[0008] In general, in one aspect, the invention features a method, apparatus, and computer-readable media for selecting records in a hierarchical set of the records having a plurality of hierarchical levels and a plurality of hierarchical depths, each of the records having a tag that is unique within the hierarchical set of the records. It comprises identifying one of the records in the hierarchical set of the records; modifying the tag; and indexing the hierarchical set of the records only once, thereby selecting one or more of the records within the hierarchical set of the records, wherein indexing the hierarchical set of the records only once comprises applying the modified tag to the hierarchical set of the record.

[0009] Particular implementations can include one or more of the following features. Each tag includes a plurality of digits; the position of each of the digits represents one of the hierarchical depths; and the value of each of the digits represents one of the hierarchical levels. Applying comprises selecting those of the records in the hierarchical set of the records having a tag that matches the key.

[0010] Advantages that can be seen in implementations of the invention include one or more of the following. Implementations of the present invention permit manipulation of a database representing an indented threaded discussion with only a single access of the database. For examples, messages can be added, deleted, and retrieved through a single database access such as a query or call.

[0011] The details of one or more implementations are set forth in the accompanying drawings and the description below. Other features will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0012] FIG. 1 shows a display of an indented threaded discussion.

[0013] FIG. 2 is a flowchart of a process for retrieving the parent of a message in an indented threaded discussion according to one implementation.

[0014] FIG. 3 is a flowchart of a process for retrieving the children of a message in an indented threaded discussion according to one implementation.

[0015] FIG. 4 is a flowchart of a process for adding a message to an indented threaded discussion according to one implementation.

[0016] The leading digit(s) of each reference numeral used in this specification indicates the number of the drawing in which the reference numeral first appears.

DETAILED DESCRIPTION

[0017] One common type of hierarchical data is an indented threaded discussion. Originally found only in Internet newsgroups, indented threaded discussions have found increasing popularity as a way to manage and display a discussion among multiple participants. Indented threaded discussion management tools are now found in consumer products such as electronic mail software programs.

[0018] FIG. 1 shows a display of an indented threaded discussion. An indented threaded discussion begins when a user posts a message 102 in a forum available to multiple users, where the posted message is not a reply to another message. Such a message is referred to herein as the "origin" message of the discussion. Any origin message constitutes the "head" of a thread. Thus message 102 is both the origin message of a discussion and the head of a thread. All replies to a message are part of the thread. Of course, a forum can have multiple discussions.

[0019] When a user posts a reply to a message, the reply is generally displayed underneath the message, and indented once. Any message having replies to it is the head of a thread. Thus a discussion can have many threads. The head of thread is indented once relative to the message to which it replies.

[0020] Message 104 is a reply to message 102. Therefore, reply 104 is displayed underneath message 102, and indented once, as shown in FIG. 1. A reply 106 to the reply 104 is indented once further. A reply 108 to the reply 106 to the reply 104 is indented once further still.

[0021] When a user posts another reply 110 to origin message 102, it is shown indented only once, and is shown beneath message 104 and its children 106 and 108. A subsequent reply 112 to origin message 102 is also shown indented once, and is shown beneath message 110. A reply 114 to message 112 is shown indented twice and beneath message 112.

[0022] A display such as the display of FIG. 1 allows a user to rapidly understand the structure of the discussion hierarchy and the place each message has in the hierarchy. The place a message has in the hierarchy can be described in terms of two dimensions of the hierarchy: level and depth. The depth of a message is the number of threads that contain the message. For example, referring to FIG. 1, the depth of message 108 is three because it is contained by the threads headed by messages 106, 104 and 102. The depth of an origin message such as message 102 is zero.

[0023] The level of a message at a particular depth represents the number of messages at that particular depth that precede that message. For example, referring to FIG. 1, the level of message 112 is three because it has a depth of one and is preceded by messages 110 and 104, both also at a depth of one. The order of precedence within each hierarchical depth can be determined by any number of factors or combinations thereof. In a threaded discussion group, the chronological order of the messages, as determined by the time and date of the posting of each message, is most commonly used. When the records in the hierarchical data set represent containers such as folders, the alphabetical order of the names of the folders can be used. In addition, the user of the display may select any field in the records, and thereby cause the contents of that field to determine the order of precedence. The level of an origin message such as message 102 is zero.

[0024] Referring again to FIG. 1, messages at a single hierarchical level, such as messages 104, 110, and 112, are shown in chronological order, while the children of a message (that is, replies to the message, replies to those replies, and so on) are shown beneath the message and above the next message at the same hierarchical level. The messages displayed to the user are generally stored in some sort of database such as a table. Each row of the table represents a message. Each message has multiple fields such as subject, author, text, date of posting, and so on. Each column represents one of the fields.

[0025] Conventional tools for managing such indented threaded discussions are notoriously inefficient. Normally either multiple rows of the table must be updated upon the creation of a new message or multiple queries have to be performed to display an indented thread. The number of rows needing updates could be as large as the number of messages in the thread.

[0026] Implementations of the present invention associate a tag with each record in a hierarchical set of records, such that the tag is unique within the hierarchical set of records. The tag is generally one of the fields of the records. Table 1 depicts a portion of a database table representing the discussion of FIG. 1 according to one implementation.

[0027]

Tag	Depth	Level	Subject
0000	0	0	Bike Rides in the Bay Area
1000	1	1	Old La Honda
1100	2	1	re: Old La Honda
1110	3	1	re: re: Old La Honda
2000	1	2	Page Mill Road
3000	1	3	The Bears Loop
3100	2	1	re: The Bears Loop

Table 1

[0028] Each tag is a number having a plurality of digits, one for each of the depths of the hierarchy. The hierarchy of Table 1 has four depths; thus each tag has four digits. In one implementation, the digits are arranged so that the most significant digit represents the first depth of the hierarchy (that is, the depth of the origin message), the next most significant digit represents the second depth of the hierarchy, and so on. Other implementations employ other arrangements of the digits.

[0029] The value of each digit represents a level in the hierarchy. Assume the hierarchy of Table 1 has four levels. Then each digit can have a values ranging from zero to three. A value of zero indicates that the message does not exist at that depth in the hierarchy. Of course, other ranges of levels and depths can be supported, as will be apparent to one skilled in the relevant art after reading this description. One implementation supports 16 levels and 16 depths. In that implementation, the tag is a 16-digit hexadecimal number represented by a "big int," a signed 64-bit number having values ranging from -2^{63} to 2^{63}.

[0030] Each tag uniquely identifies the position of its record in the hierarchy. For example, the tag "3100" shows that the record is the first child of the third child of the origin message.

[0031] One advantage of the disclosed tag is that the tag of a record may be determined quickly and easily, without indexing the database, by modifying the tag of a related record.

[0032] FIG. 2 is a flowchart of a process 200 for retrieving the parent of a message in an indented threaded discussion according to one implementation. Process 200 receives a command from a user (step 202). The command identifies message 108 and requests retrieving the parent of message 108. Process 200 modifies the tag of message 108 to produce a key. To do this, process 200 first selects the digit corresponding to the hierarchical depth of message 108 (step 204). Referring to Table 1, message 108 has a depth of three. Therefore, process 200 selects the third most significant digit of the tag of message 108. The depth of each message need not be stored in the table, but can be determined in real time, with knowledge of the hierarchical depth represented by each digit. Where tags have digits arranged according to increasing depths in the hierarchy, such as the tags of Table 1, one can select the digit corresponding to the hierarchical depth of a message by simply select the least significant non-zero digit.

[0033] Process 200 then sets the value of the selected digit to a null value (step 206). In the tags of Table 1, the null value is zero. Of course, other values can be selected as the null value. The resulting key is "1100", which is the tag of message 106, the parent of message 108. Process 200 then indexes the database only once by applying the key to the database (step 208), thereby selecting the record for message 106. Process 200 then retrieves message 106 (step 210) and displays message 106 to the user (step 212).

[0034] FIG. 3 is a flowchart of a process 300 for retrieving the children of a message in an indented threaded discussion according to one implementation. Process 300 receives a command from a user (step 302). The command identifies message 104 and requests retrieving all of the messages in the thread for which message 104 is the head. These messages include the children of message 104, the children of the children of message 104, and so on.

[0035] Process 300 modifies the tag of message 104 to produce a key. To do this, process 300 first selects the digit corresponding to the hierarchical depth of message 104 (step 304). Referring to Table 1, message 104 has a depth of one. Therefore, process 300 selects the most significant digit of the tag of message 104. Process 300 then sets the value of each digit corresponding to a hierarchical depth below the hierarchical depth of the selected digit to a wildcard value (step 308). The resulting key is "1XXX", where "X" represents a wildcard value that can match any value.

[0036] In one implementation, applying a key to the database selects all records in the database having tags that match the key. Process 300 indexes the database only once by applying the key to the database (step 310), thereby selecting the records for the messages in the thread headed by message 104. Referring to Table 1, the key "1XXX" matches the tags for records corresponding to messages 106 and 108. Process 300 then retrieves messages 106 and 108 (step 312) and displays messages 106 and 108 to the user (step 314).

[0037] Process 300 can also be used to delete a message. When a message is deleted, so are all of its children. Process 300 selects a message and all of its children. The records corresponding to the selected messages are then deleted.

[0038] FIG. 4 is a flowchart of a process 400 for adding a message to an indented threaded discussion according to one implementation. Process 400 receives a command from a user (step 402). The command identifies message 114 and requests replying to message 114. Process 400 modifies the tag of message 114 to produce a key. To do this, process 400 first selects the digit corresponding to the hierarchical depth immediately below the hierarchical depth of message 114 (step 404). Referring to Table 1, message 114 has a depth of two. Therefore, process 400 selects the third most significant digit of the tag of message 104. Process 400 then increments the value of selected digit (step 406). The resulting key is "3110". Process 400 then indexes the database only once by adding a record to the database having the key as its tag (step 408). The record corresponds to the new message.

[0039] The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

[0040] A number of implementations of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other implementations are within the scope of the following claims.